Gremlin Input Coprocessor

Table of Contents

[Coprocessor 2](#_Toc315601193)

[Inputs 2](#_Toc315601194)

[PS/2 Port for Mouse 2](#_Toc315601195)

[PS/2 Port for Keyboard 2](#_Toc315601196)

[Outputs 2](#_Toc315601197)

[SPI bus (output only) 2](#_Toc315601198)

[Gremlin 2](#_Toc315601199)

[Inputs 3](#_Toc315601200)

[SPI Bus (Input only) 3](#_Toc315601201)

[Outputs 3](#_Toc315601202)

# Overview

This document describes one possible solution for handling I/O on the Gremlin computer.

The objective of this design is to offload the majority of the workload to a co-processor, so as to minimize the amount of implementation within the FPGA chip.

The FPGA chip contains a lot of SDRAM memory. Some of this memory can be set aside to act as a keyboard and mouse input buffer.

The Keyboard buffer would contain up-to 256 bits (32 bytes) of data. Each bit would represent an UP (0) or DOWN (1) state for each key on a PS/2 keyboard. It would also contain 1 byte of writable data, which would control the keyboard LED’s.

The Mouse buffer would contain 3 bytes of data (X, Y, and button state to start with, and possibly scroll-wheel and additional buttons in the future if the buffer length is increased).

All data is read-only, from the perspective of the Gremlin.

## Packet Structure A

The following table describes the complete data packet to be exchanged between coprocessor and FPGA.

The R/W column is from the perspective of the application which is running on the FPGA.

|  |  |  |
| --- | --- | --- |
| Index | R/W | Description |
| 0..27 | READ | Keyboard button states |
| 28 | READ | Mouse Button State |
| 29 | READ | Mouse X position |
| 30 | READ | Mouse Y position |
| 31 | WRITE | Keyboard LED states |

## Packet Structure B

In this scenario, the coprocessor is the master, and the FPGA is the slave.

The coprocessor will send a packet of data whenever it detects a keypress or mouse data change, and can limit the maximum throughput to some arbitrary speed, such as 60 times per second. Limiting the amount of traffic limits the amount of time spent locking the CPU bus, at the FPGA side. We’ll also limit the amount of data per packet as well.

### BYTE packet

This packet contains one index byte, and one data byte. When the FPGA receives this packet, it writes the data at the specified index.

READ\_BYTE Packet

This packet contains one index byte, and one NULL byte. When the FPGA deceives this packet, it ignores the null data, and injects

## Programming Interface (Game Logic)

# Coprocessor

## Inputs

### PS/2 Port for Mouse

### PS/2 Port for Keyboard

## Outputs

### SPI bus (output only)

# Gremlin

## Inputs

### SPI Bus (Input only)

## Outputs

(none)